Imprecise Computation Task Mapping on Multi-Core Wireless Sensor Networks

Lei Mo, Angeliki Kritikakou, Olivier Sentieys

Imprecise Computation Task Mapping on Multi-Core Wireless Sensor Networks

Lei Mo, Angeliki Kritikakou, and Olivier Sentieys

Synonyms

Imprecise Computations, Multicore Architectures, Task Mapping.

Definitions

Imprecise Computation (IC) model (12) considers that a task can be logically decomposed into a *mandatory* subtask and an *optional* subtask. The mandatory subtask should be completed before the task's deadline in order to generate the minimum acceptable Quality of Service (QoS). The optional subtask is to be executed after the mandatory subtask, and still before the deadline, if there are resources in the system that are free to execute the optional subtask. The longer the optional subtask is executed, the better is the QoS of the obtained result.

Historical Background

Wireless Sensor Networks (WSNs) consist of a large number of wireless sensor nodes that have the capability to take various measurements of their environment. These measurements could be temperature, sound, vibration, pressure or motion etc. *Energy efficiency* and *real-time execution* are critical and challenging issues for the design of WSNs systems (15). This is because:

Lei Mo, Angeliki Kritikakou, and Olivier Sentieys
Univ Rennes, INRIA, IRISA, CNRS, 35042 Rennes Cedex
e-mail: lei.mo@inria.fr, angeliki.kritikakou@irisa.fr, olivier.sentieys@irisa.fr

1. Most of the sensor nodes have energy constraints, as they are powered by a battery that has a limited energy budget.
2. Real-time responsiveness is required by many WSN applications, e.g., mobile target tracking. Missing of the task deadline can cause serious results.

In the past, many research works have focused on optimizing the energy efficiency and real-time execution of applications on WSNs, e.g., data aggregation (5), topology control (6), sink mobility (19) and delay-aware routing (9). These methods improved significantly the energy efficiency and real-time execution of the WSNs. However, further improvements exist:

1. Most of the past works focused on the software optimization methods without including the hardware optimization strategies.
2. Most of the past works focused on reducing the energy consumption of the wireless transmission, but ignored the energy consumption of the processor modules.

However, processing and transmitting large amount of sensed data in real-time applications exceeds the capabilities of traditional WSNs. Since *single-core* embedded sensor nodes will soon be unable to meet the increasing requirements of data intensive applications, e.g., Wireless Multimedia Sensor Networks (WMSNs), the next generation sensor nodes must possess enhanced computation and communication capabilities, i.e., *multi-core* embedded sensor nodes (18). For instance, the fire detection WMSN node MiLive-v2 in (13) is equipped with three types of processors:

1. A 8-bit low-power AVR processor (ATmega128rfa1), which can be used to run simple tasks.
2. A 32-bit powerful ARM processor (ARM1176JZF), which can be used to run more complicated tasks.
3. A Digital Signal Processor (DSP) unit, which can be specially used to perform image processing.

The requirements of WSNs are usually high performance computation, low energy consumption and low task execution delay. The efficient mapping of the tasks on multi-core sensor nodes (13) is required to balance these *contradictory* requirements. It can simultaneously optimize *task allocation* (on which processor each task is executed) and *task scheduling* (when each task starts and ends its execution). On the one hand, in order to map a set of applications on multi-core system, the applications need to be partitioned (parallelized), whenever possible, into multiple tasks that can be executed concurrently on different processors. Therefore, task execution delay can be further reduced. On the other hand, multi-core systems have better energy savings than traditional single-core systems in two ways:

1. Multi-core embedded sensor nodes can extract the desired information from the sensed data and process this information. It leads to reduction in the data transmission volume sent to the sink node (base station). By replacing a large percentage of communication with in-network computation, the multi-core system realizes large energy savings that increase the sensor network's overall lifetime.

2. A multi-core embedded sensor node allows the computations to be split across multiple processors, while running each processor at a lower voltage and frequency, as compared to a single-core system, which results in energy savings. Utilizing a single-core embedded sensor node for information processing in data intensive applications demands the sensor node run at a high voltage and frequency in order to meet the application's real-time requirements. It will increase the power dissipation of the processor.

The tradeoff between energy consumption and performance has been extensively studied using the following methods:

1. *Dynamic Voltage and Frequency Scaling* (DVFS): by lowering the supply voltage, quadratic savings in dynamic energy consumption can be achieved, while the performance is approximately degraded in linear manner.
2. *Dynamic Power Management* (DPM): by exploring idle intervals of the processors and switching off a processor, when the idle interval is longer than a certain threshold, which is called break-even time.

Existing works that focus on energy-aware task mapping problem aim at minimizing the energy consumption under resource and application constraints.

1. When the voltage level is *discrete*, the task mapping problem is usually formulated as Integer Programming (IP), e.g., (14; 22; 11). To efficiently solve the IP problem, a hybrid Genetic Algorithm (GA) is proposed in (14), a polynomial-time two-step heuristic is designed in (22), and the IP problem is relaxed to a Linear Programming (LP) in (11). Combining DVFS and DPM, a Mixed-Integer Linear Programming (MILP)-based task mapping problem is considered in (3) and the problem is solved by CPLEX solver.
2. When the voltage level is *continuous*, a convex task mapping problem is formulated in (8) and the problem can be solved by using polynomial-time methods. In (10), Mixed-Integer Non-Linear Programming (MINLP) is used to formulate the task mapping problem. The problem is relaxed to an MILP by linear approximation and solved by Branch and Bound (B&B) method.

As long as the resource and application constraints (e.g., task deadline) allow, the methods with DVFS or DPM achieve significant energy reductions compared to the basic methods, where no DVFS or DPM is applied.

Key Applications

In several application domains, an approximate–but still in time–result is preferable than an exact result obtained too late. For example, in audio and video streaming, frames with a lower quality are better than missing frames. In radar tracking, an estimation of target's location in time is better than an accurate location arriving too late. In control loops, an approximate result produced by a control scheme is more preferable as long as the controlled system, e.g., indoor temperature control system,

remains stable. This type of applications can be modeled as *Imprecise Computation* (IC) tasks. The more cycles of the optional subtasks are executed, the higher is the generated QoS. An interesting question is how should the function of QoS be defined in order to represent realistic application areas:

1. A *linear* function (25; 23; 26; 21; 16; 17) models the case where the QoS of the overall system increases uniformly during the optional subtask execution.
2. A *concave* function (2; 24; 4; 20) addresses the case where the increase of QoS exhibits a continuously nondecreasing rate as the optional subtask execution goes on.

Linear and general concave functions are considered as the most realistic and typical QoS representation in the literature, as they adequately capture the behavior of many application areas, such as image and speech processing, control engineering, and automatic target recognition (1; 2; 12).

Foundations

The three constraints *energy*, *deadline*, and *QoS* play an important roles in the QoS-aware task mapping. This is different from the energy-aware task mapping, where no exploration of the QoS improvement through the optional subtasks adjustment take place. The QoS-aware task mapping problem uses the IC task model and has a goal to maximize the QoS under a set of real-time and energy supply constraints, e.g., (2; 4; 24; 25; 20; 23; 7; 26; 21; 16; 17). Specifically, QoS-aware task mapping determines on which processor should the task be executed (task-to-processor allocation), its frequency (frequency-to-task assignment), the start time (task scheduling), and the end time (optional subtask adjustment) of each task, such that the system QoS is maximized, while meeting the energy supply and the task deadlines. Usually, the task-to-processor allocation and frequency-to-task assignment decisions are *binary* variables, while the task scheduling and optional subtask adjustment decisions can be either *integer* (24; 25; 23; 7; 26; 21) or *continuous* (2; 4; 20; 16; 17) variables. Based on different constraints and variables introduced into the problem formulation, the variables may be coupled with each other nonlinearly, i.e., MINLP. However, it is not desirable since solving MINLP is much more complex than solving MILP. In order to linearize the nonlinear items, the common methods include:

1. Linear approximation (10);
2. Variables replacement (3; 17).

The QoS-aware task mapping problem is similar to *Quadratic Assignment Problem*, a well-known NP-hard problem. Therefore, finding an optimal solution satisfying all the given constraints (e.g., energy efficiency, deadline, QoS, task dependency, and DVFS) is very difficult and time consuming (17). Existing works usually focus on different contexts, as summarized in Table 1. The platforms considered

in (4; 24; 20) are single-core platforms, i.e., the task allocation problem is not taken into account. Although the works in (2; 25; 23; 7; 26; 21; 16) target at multi-core platforms, some assumptions are made during the problem formulation, e.g., the task-to-processor allocation is fixed and given in advance for all the tasks in (25; 23), the tasks considered in (7) are independent, and in (2; 26; 21; 16) each processor has a predefined frequency. The methods used to solve the aforementioned problems can be classified into two main classes:

1. The first class includes the methods based on heuristics, e.g., (24; 25; 23; 7; 26; 21).
2. The second class includes the methods that always produce an optimal solution, e.g., (2; 4; 20; 16; 17).

Although heuristic methods can find feasible solutions in a short amount of time, they do not provide the bounds on the solution quality. When new assumptions or constraints are taken into account they must be redeveloped, and, thus, they are not always easily extensible.

Table 1 Task Mapping Method

		Energy-aware						*QoS-aware*										
		(3)	(8)	(10)	(11)	(14)	(22)	(2)	(4)	(7)	(23)	(16)	(21)	(24)	(25)	(26)	(20)	(17)
Variables	Frequency-to-task	√	√	√		√	√		√	√	√			√	√			√
	Task-to-processor	√		√	√	√		√		√		√	√			√		√
	Task adjustment							√	√	√	√	√	√	√	√	√	√	√
Objective	Max. QoS							√	√	√	√	√	√	√	√	√	√	√
	Min. Energy	√	√	√	√	√	√			√								
Constraints	Real-time	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
	Energy								√		√	√	√	√	√	√		√
Solution	Optimal	√	√					√	√			√					√	√
	Non-optimal			√	√	√	√			√	√		√	√	√	√		√

The heuristic methods mentioned above usually adopted a multi-step optimization, i.e., to decouple the variables and to determine their values in sequence. For instance, a two-step heuristic called Adaptive Task Allocation (ATA) is proposed in (26). The aim of the first step is to find a task-to-processor allocation, such that the energy consumption is minimized. With the given task allocation decision, the energy consumed to execute one task is proportional to the length of its optional subtask. Based on the given task allocation decision, the aim of the second step is to adjust the optional subtasks so as to maximize OoS under the energy supply constraint. On the other hand, in order to find the optimal solution, the common methods include:

1. Convex optimization (2; 4);
2. Benders Decomposition (BD) (16; 17).

The structure of BD is shown in Fig. 1. Instead of solving the binary and the continuous variables of MILP problem simultaneously, BD technique decomposes the original problem into two smaller problems with less variables and constraints: an ILP-based *Master Problem* (MP) and a LP-based *Slave Problem* (SP), and then solves them by utilizing the solution of one in the other. By doing so, the computation time

can be significantly reduced. In each iteration, the current MP is solved to determine

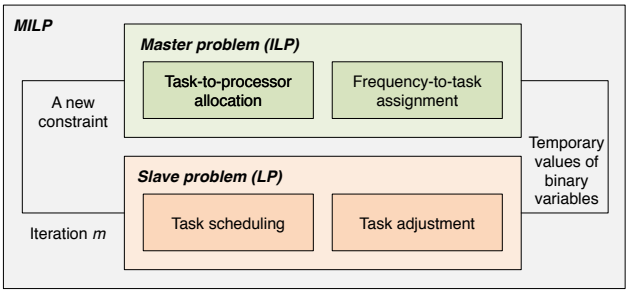

Fig. 1 The structure of Benders decomposition.

a lower bound for the original problem along with the temporary values of the binary variables. And then the SP is solved to obtain an upper bound by utilizing the MP solution. The bounds are updated if the stopping criterion, i.e., the gap between the upper and lower bounds is smaller than a predefined threshold, is not met, and a new constraint (Benders cut) is generated using the SP solution and is added to MP in next iteration.

References

1. Aydin, H., Melhem, R., Mosse, D.: Tolerating faults while maximizing reward. In: IEEE Euromicro Conference on Real-Time Systems (EMRTS), pp. 219–226 (2000)
2. Aydin, H., Melhem, R., Mosse, D., Mejia-Alvarez, P.: Optimal reward-based scheduling for periodic real-time tasks. IEEE Trans. Comput. **50**(2), 111–130 (2001)
3. Chen, G., Huang, K., Knoll, A.: Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination. ACM Trans. Embed. Comput. Syst. **13**(3), 111:1–111:21 (2014)
4. Cortes, L.A., Eles, P., Peng, Z.: Quasi-static assignment of voltages and optional cycles in imprecise computation systems with energy considerations. IEEE Trans. Very Large Scale Integr. Syst. **14**(10), 1117–1129 (2006)
5. Dhasian, H.R., Balasubramanian, P.: Survey of data aggregation techniques using soft computing in wireless sensor networks. IET Inform. Secu. **7**(4), 336–342 (2013)
6. Esch, J.: A survey on topology control in wireless sensor networks: taxonomy, comparative study, and open issues. Proc. IEEE **101**(12), 2534–2537 (2013)
7. Isabel, M., Javier, O., Rodrigo, S., Paula, Z.: Energy-aware scheduling mandatory/optional tasks in multicore real-time systems. Intl. Trans. in Op. Res. **24**(12), 173–198 (2017)
8. Kong, F., Yi, W., Deng, Q.: Energy-efficient scheduling of real-time tasks on cluster-based multicores. In: ACM/IEEE Design, Automation Test in Europe (DATE), pp. 1–6 (2011)
9. Lai, X., Ji, X., Zhou, X., Chen, L.: Energy efficient link-delay aware routing in wireless sensor networks. IEEE Sensors J. **18**(2), 837–848 (2018)
10. Leung, L.F., Tsui, C.Y., Ki, W.H.: Simultaneous task allocation, scheduling and voltage assignment for multiple-processors-core systems using mixed integer nonlinear programming. In: IEEE International Symposium on Circuits and Systems (ISCAS), pp. 309–312 (2003)

11. Li, D., Wu, J.: Minimizing energy consumption for frame-based tasks on heterogeneous multiprocessor platforms. IEEE Trans. Parallel Distrib. Syst. **26**(3), 810–823 (2015)
12. Liu, J.W.S., Shih, W.K., Lin, K.J., Bettati, R., Chung, J.Y.: Imprecise computations. Proc. IEEE **82**(1), 83–94 (1994)
13. Liu, X., Zhou, H., Xiang, J., Xiong, S., Hou, K.M., de Vaulx, C., Wang, H., Shen, T., Wang, Q.: Energy and delay optimization of heterogeneous multicore wireless multimedia sensor nodes by adaptive genetic-simulated annealing algorithm. Wirel. Commun. Mob. Comput. **2018**, 1–13 (2018)
14. Mahmood, A., Khan, S.A., Albalooshi, F., Awwad, N.: Energy-aware real-time task scheduling in multiprocessor systems using a hybrid genetic algorithm. Electron. **6**(2) (2017)
15. Mo, L., Cao, X., Song, Y., Kritikakou, A.: Distributed node coordination for real-time energy-constrained control in wireless sensor and actuator networks. IEEE Internet Things J. pp. 1–12 (2018). DOI 10.1109/JIOT.2018.2839030
16. Mo, L., Kritikakou, A., Sentieys, O.: Decomposed task mapping to maximize QoS in energy-constrained real-time multicores. In: IEEE International Conference on Computer Design (ICCD), pp. 493–500 (2017)
17. Mo, L., Kritikakou, A., Sentieys, O.: Controllable QoS for imprecise computation tasks on DVFS multicores with time and energy constraints. IEEE J. Emerg. Sel. Topics Circuits Syst. pp. 1–14 (2018). DOI 10.1109/JETCAS.2018.2852005
18. Munir, A., Gordon-Ross, A., Ranka, S.: Multi-core embedded wireless sensor networks: architecture and applications. IEEE Trans. Parallel Distrib. Syst. **25**(6), 1553–1562 (2014)
19. Silva, R., Silva, J.S., Boavida, F.: Mobility in wireless sensor networks: a survey and proposal. Comput Commun **52**, 1–20 (2014)
20. Tchamgoue, G.M., Kim, K.H., Jun, Y.K., Lee, W.Y.: Compositional real-time scheduling framework for periodic reward-based task model. J. Syst. Softw. **86**(6), 1712–1724 (2013)
21. Wei, T., Zhou, J., Cao, K., Cong, P., et al.: Cost-constrained QoS optimization for approximate computation real-time tasks in heterogeneous MPSoCs. IEEE Trans. Comput.-Aided Design Integr. Circuits Syst. **1**(1), 1–14 (2017)
22. Xu, H., Kong, F., Deng, Q.: Energy minimizing for parallel real-time tasks based on level-packing. In: IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), pp. 98–103 (2012)
23. Yu, H., Ha, Y., Veeravalli, B.: Quality-driven dynamic scheduling for real-time adaptive applications on multiprocessor systems. IEEE Trans. Comput **62**(10), 2026–2040 (2013)
24. Yu, H., Veeravalli, B., Ha, Y.: Dynamic scheduling of imprecise-computation tasks in maximizing QoS under energy constraints for embedded systems. In: IEEE Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 452–455 (2008)
25. Yu, H., Veeravalli, B., Ha, Y., Luo, S.: Dynamic scheduling of imprecise-computation tasks on real-time embedded multiprocessors. In: IEEE International Conference on Computational Science and Engineering (CSE), pp. 770–777 (2013)
26. Zhou, J., Yan, J., Wei, T., Chen, M., Hu, X.S.: Energy-adaptive scheduling of imprecise computation tasks for QoS optimization in real-time MPSoC systems. In: ACM/IEEE Design, Automation Test in Europe (DATE), pp. 1402–1407 (2017)